5.3 A 4320MIPS Four-Processor Core SMP/AMP with Individually Managed Clock Frequency for Low Power Consumption

Yutaka Yoshida[1], Tatsuya Kamei[1], Kiyoshi Hayase[1], Shinichi Shibahara[1], Osamu Nishii[1], Toshihiro Hattori[1], Atsushi Hasegawa[1], Masashi Takada[2], Naohiko Irie[2], Kunio Uchiyama[2], Toshihiko Odaka[2], Kiwamu Takada[3], Keiji Kimura[4], Hironori Kasahara[4]

[1]Renesas Technology Tokyo, Japan, [2]Hitachi, Tokyo, Japan
[3]Hitachi ULSI, Tokyo, Japan, [4]Waseda University, Tokyo, Japan

A 4320MIPS four-processor-core SoC is designed for power-efficient high-performance embedded applications where power consumption, small size and high performance are critical concerns. Each processor core can operate with a different frequency, varying dynamically, and including stopping the clock, while keeping data caches coherent. These features maintain maximum processing performance and reduce the average operating power. Each processor core includes a CPU, an FPU, 32KB 4-way set-associative instruction and data caches, a 4-entry instruction TLB, a 64-entry unified TLB, 8KB and 16KB local RAM (ILRAM, OLRAM) and 128KB user RAM (URAM). Figure 5.3.1 shows a block diagram of the four-processor SoC, which includes a snoop controller (SNC) to maintain data cache coherency among processor cores, memory interfaces such as DDR2-SDRAM and SRAM, a PCI Express interface, some hardware IPs for various types of data processing, and other peripheral modules. The hardware IPs are connected with an on-chip system bus (SHwy) which is a packet-based split-transaction bus [1]. The processor cores and SNC are connected with a snoop bus that is separate from SHwy, to avoid increasing bus traffic on SHwy.

This system meets the following needs of embedded application: (i) standard common on-chip-bus-based SoC integration, (ii) good adaptation to both SMP and AMP, (iii) dynamic power management, cooperating with software such as an automatic parallelizing compiler [2], [3], that can control the power of processors, adapting to the processing load.

This chip supports both SMP with standard MESI cache coherency protocol and AMP that can avoid coherency tag check cycle overhead. In AMP mode, 152KB local memories for each processor can be used for data sharing among processor cores. The four cores can run with different frequencies , which are controlled dynamically with short switching time. Data cache coherency is kept between multiple processors with different frequencies, including the case in which the processor frequencies are lower than the on-chip system bus clock (SCLK). The assumed usage of this SoC is with fixed SCLK and varying CPU frequency according to CPU workloads.

To be able to change frequency of each processor-core individually while keeping a data cache coherency, we use four schemes as follows:

- Each core has a clock divider to change clock frequency individually.
- A handshake protocol is executed for the target processor core before changing the clock frequency to avoid a conflict between bus access and the clock change,
- The cores support various clock frequency ratios between CPU and SCLK, including the case in which the CPU frequency is lower than that of SCLK,
- Each processor core has a light-sleep mode to stop the CPU clock while keeping a data cache coherent.

Figure 5.3.2 shows the clock distribution, in which the clock gating circuitry includes a clock divider for processor cores and modules. The clock gating circuitry uses standard cell design and consists of a NOR element, AND element, and a latch. The latch prevents a noise hazard from the clock line and optimizes setup time from the enable signal. In a previous design, the NOR element has two gates: a NOR gate and a clocked inverter. However, this work has only one gate, a clocked NOR gate, which decreases setup time and reduces cell area. The processor-core ICLK runs up to 600MHz and SCLK runs up to 300MHz. These clocks are generated by the GCPG and distributed commonly for each processor core. Both ICLK and SCLK are programmable by the frequency-control register in the GCPG, as seen in Cases G1, G2 and Cases S1, S2 of Fig. 5.3.2. An internal processor-core clock (ICK) is generated from ICLK by each clock divider (C0, C1, C2 and C3) for each processor core individually such as Case i1, i2, i3 and i4. When the frequency control register of a target processor core is set, the LCPG executes a handshake sequence to keep data transfer consistency for the target processor core, so that it can prevent stalling other processor cores and maintain the data-transfer performance of some IPs on SHwy. The system also supports *light-sleep mode*, during which ICK is provided only to the data cache to keep data cache coherency, but not to the CPUs, instruction cache or other modules . This mode reduces the power consumption for SMP use of the system.

Figure 5.3.3 shows the coherency functional block and an example of the operation of a single transaction from shared status (S) to modified status (M) in the standard MESI protocol. Steps 1 to 7 in the figure show the sequence of operation of a transaction. SNC has a duplicated address array (DAA) that is a copy of the data cache address tag of processor core. The data cache of each core is accessed using ICK timing, and the DAA is accessed at SCLK timing. Each processor core has a data synchronization circuit and uses a handshake protocol to keep exact execution on the snoop-bus. A snoop acknowledge (step 4) is returned to Core #0 before the invalidate acknowledge (step 7) of Core #1 to hide the overheads of Core #1. If the invalidation operation of Core #1 is incomplete, Core #0 is stalled by step 4.

Figure 5.3.4 shows the coherency overhead cycles. Core #0 is a request source and runs at 600MHz, Core #1 has a possibility of being the owner of a target tag and runs at 600MHz or 150MHz. Core #0 issues a request to the SNC in the case of cache miss or write hit to a shared tag. The SNC has a DAA and it can be checked in the SNC without access to Core #1. When the target tag is not in the DAA or the target tag is in shared status (S) in Core #1, the clock frequency of Core #1 has no influence on overhead, but if the target tag is in exclusive status (E) or modified status (M), it takes a few overhead cycles, depending on the clock frequency of Core #1, to maintain coherency.

Figure 5.3.5 shows the estimated power consumption of a benchmark for four different cases. An instruction cycle simulator determines the required cycle count of an MPEG2 decode benchmark with 4 CPUs in SMP mode,. The power consumption is calculated as the product of the cycle time derived from simulation and the power consumption of the processor core, which is calculated in advance for the three cases of typical active use, waiting in the loop state, and light-sleep mode. The figure shows the average power consumption of each core: Core #0, Core #1, Core #2, and Core #3. We show four cases of processor use as follows: (1) all processors run at 600MHz and some CPUs are waiting for completion of all other tasks in the loop state (IDLE), (2) all processors run at 600MHz and some CPUs are in light-sleep mode after finishing their tasks, (3) Core #0 and Core #1 run at 600MHz, Core#3 runs at 300MHz and Core#4 runs at 150MHz and some CPUs become IDLE, and, (4) CPUs run at 600MHz, 600MHz, 300MHz and 150MHz with some CPUs in light-sleep mode after finishing their tasks . Case (1) is the base case and draws 1.4W. Case (2) reduces the average power consumption to 88% using light-sleep mode. Case (3) reduces the average power consumption to 83% using individual clock frequency control. Finally, case (4) reduces the average power consumption to 80% using both light-sleep mode and individual clock frequency.

References:
[1] K. Uchiyama, et al., "Embedded Processor Core with 64-Bit Architecture and Its System-On-Chip Integration for Digital Consumer," *IEICE*, pp. 139-149, Feb. 2001.
[2] J. Shirako, et al., "Compiler Control Power Saving Scheme for Multi Core Processors," Proc. of The 18th International Workshop on Languages and Compilers for Parallel Computing (LCPC2005), Oct., 2005.
[3] K. Kimura, Y. Wada, H. Nakano et al., "Multigrain Parallel Processing on Compiler Cooperative Multiprocessor," Workshop on Interaction Between Compilers and Computer Architectures, pp. 11-20, Feb., 2005.

 1-4244-0852-0/07/$25.00

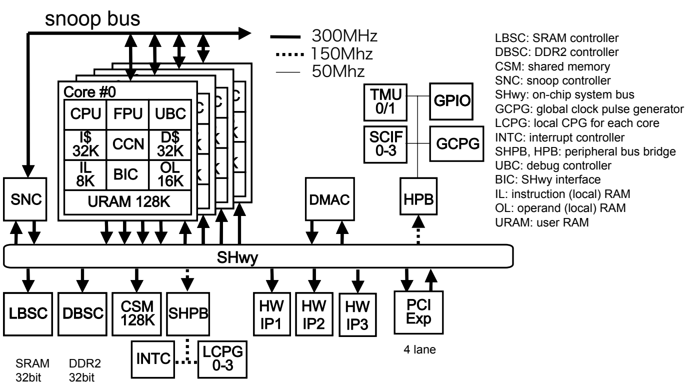

Figure 5.3.1: Block diagram of the processor.

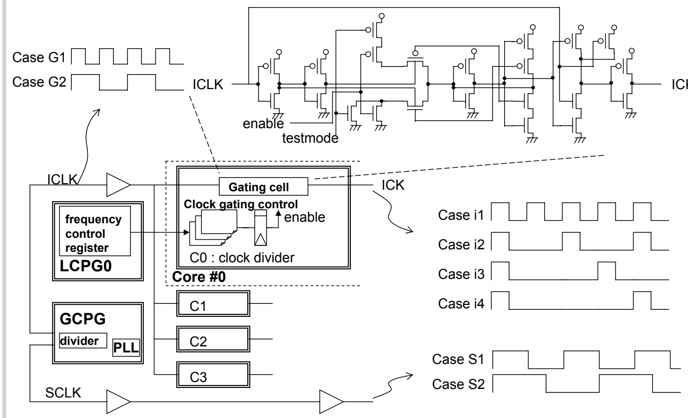

Figure 5.3.2: Clock gating circuit and clock distribution.

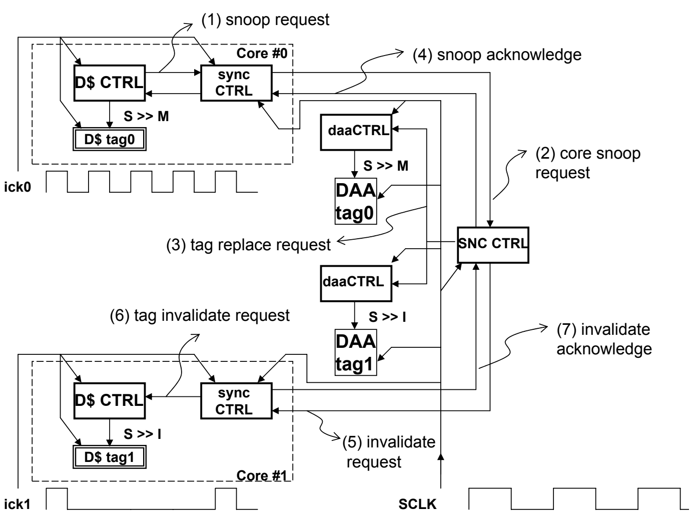

Figure 5.3.3: Coherency functional block and example of operation.

Core #0 status	target tag status		overhead (SCLK cycles)	
			Core #1=600MHz (*1	Core #1=150MHz (*1
Read hit	shared / not shared		0	same
Write hit	not shared		0	same
	shared		4 (*2	same
Read/Write miss	DAA miss:Invalid (I)		5	same
	DAA hit	Shared (S)	10 (*2	same
		Exclusive (E)	10	1.9 times
		Modified (M)	13	1.9 times

*1: Core #0 runs at 600MHz and SNC runs at 300MHz
*2: case of single transaction

Figure 5.3.4: Coherency overhead cycles.

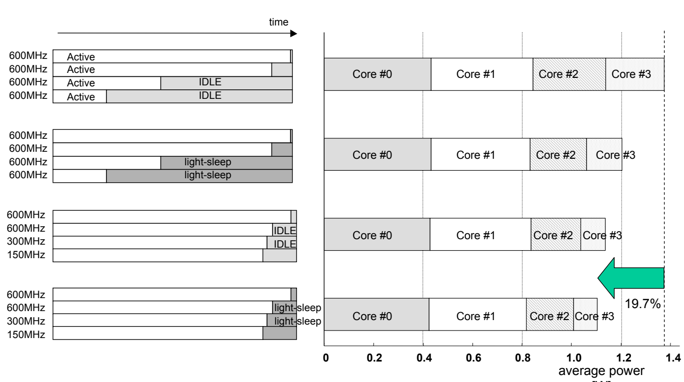

Figure 5.3.5: Result of estimation.

Process Technology	90nm, 8-layer, triple-Vth, CMOS
Chip Size	97.6mm² (9.88mm x 9.88mm)
Supply Voltage	1.0V (internal), 1.8/3.3V (I/O)
Clock Frequency	600MHz
CPU Performance	4320 MIPS (Dhrystone 2.1)
FPU Performance	16.8 GFLOPS
I/D Cache	32KB 4way set-associative (each)
ILRAM/OLRAM	8KB/16KB (each CPU)
User Memory	128KB (each CPU)
Package	FCBGA 554pin, 29mm x 29mm

Figure 5.3.6: Chip specification.

Continued on Page 590

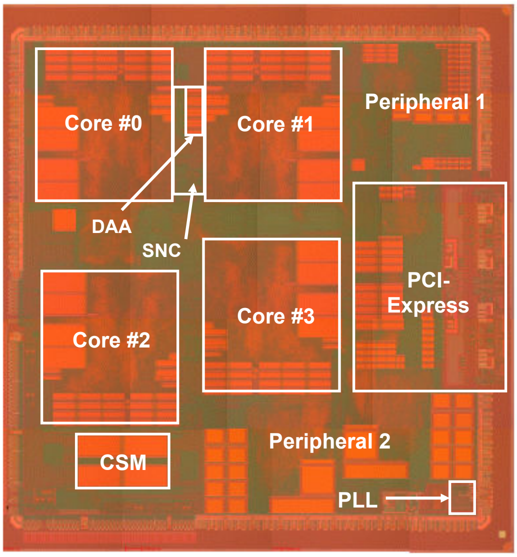

Figure 5.3.7: Die photograph.

